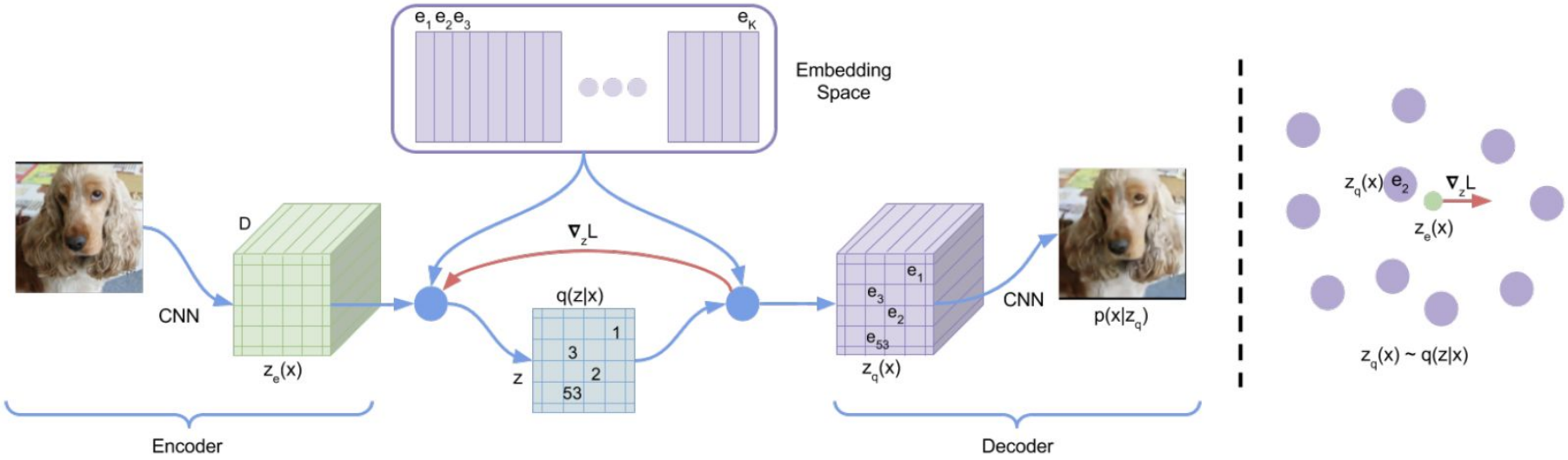
# Generating Diverse High-Fidelity Images with VQ-VAE-2

DeepMind

# Background (VQ-VAE)

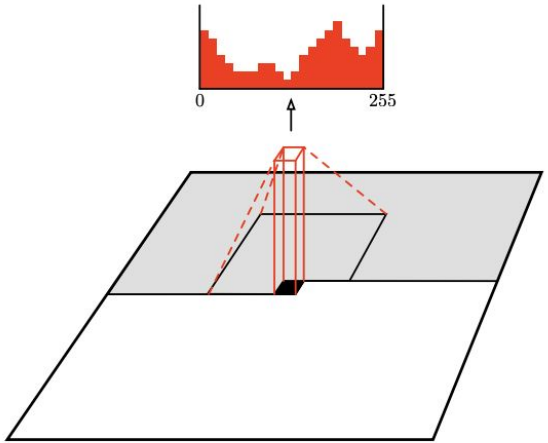


# Background (VQ-VAE) (cont.)

$$\mathcal{L}(\mathbf{x}, D(\mathbf{e})) = ||\mathbf{x} - D(\mathbf{e})||_2^2 + ||sg[E(\mathbf{x})] - \mathbf{e}||_2^2 + \beta||sg[\mathbf{e}] - E(\mathbf{x})||_2^2 \tag{2}$$

$$N_i^{(t)} := N_i^{(t-1)} * \gamma + n_i^{(t)}(1-\gamma), \quad m_i^{(t)} := m_i^{(t-1)} * \gamma + \sum_j^{n_i^{(t)}} E(x)_{i,j}^{(t)}(1-\gamma), \quad e_i^{(t)} := \frac{m_i^{(t)}}{N_i^{(t)}}$$

# Conditional PixelCNN




| | | | | |
|---|---|---|---|---|
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 1 | 1 | 1 |
| 1 | 1 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |
| 0 | 0 | 0 | 0 | 0 |

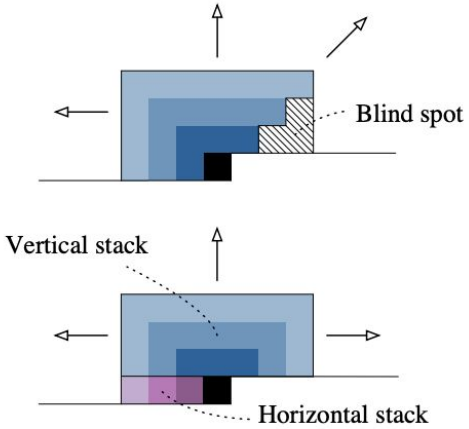



Figure 1: **Left**: A visualization of the PixelCNN that maps a neighborhood of pixels to prediction for the next pixel. To generate pixel $x_i$ the model can only condition on the previously generated pixels $x_1, \ldots x_{i-1}$. **Middle**: an example matrix that is used to mask the 5x5 filters to make sure the model cannot read pixels below (or strictly to the right) of the current pixel to make its predictions. **Right**: Top: PixelCNNs have a *blind spot* in the receptive field that can not be used to make predictions. Bottom: Two convolutional stacks (blue and purple) allow to capture the whole receptive field.

# Conditional PixelCNN (cont.)

Given a high-level image description represented as a latent vector $\mathbf{h}$, we seek to model the conditional distribution $p(\mathbf{x}|\mathbf{h})$ of images suiting this description. Formally the conditional PixelCNN models the following distribution:

$$p(\mathbf{x}|\mathbf{h}) = \prod_{i=1}^{n^2} p(x_i|x_1, ..., x_{i-1}, \mathbf{h}). \tag{3}$$

We model the conditional distribution by adding terms that depend on $\mathbf{h}$ to the activations before the nonlinearities in Equation 2, which now becomes:

$$\mathbf{y} = \tanh(W_{k,f} * \mathbf{x} + V_{k,f}^T \mathbf{h}) \odot \sigma(W_{k,g} * \mathbf{x} + V_{k,g}^T \mathbf{h}), \tag{4}$$

# Method

**Algorithm 1** VQ-VAE training (stage 1)

**Require:** Functions $E_{top}$ , $E_{bottom}$, $D$, $\mathbf{x}$ (batch of training images)

$\mathbf{h}_{top} \leftarrow E_{top}(\mathbf{x})$

▷ quantize with top codebook eq 1

$\mathbf{e}_{top} \leftarrow Quantize(\mathbf{h}_{top})$

$\mathbf{h}_{bottom} \leftarrow E_{bottom}(\mathbf{x}, \mathbf{e}_{top})$

▷ quantize with bottom codebook eq 1

$\mathbf{e}_{bottom} \leftarrow Quantize(\mathbf{h}_{bottom})$

$\hat{\mathbf{x}} \leftarrow D(\mathbf{e}_{top}, \mathbf{e}_{bottom})$

▷ Loss according to eq 2

$\theta \leftarrow Update(\mathcal{L}(\mathbf{x}, \hat{\mathbf{x}}))$

**Algorithm 2** Prior training (stage 2)

$\mathbf{T}_{top}, \mathbf{T}_{bottom} \leftarrow \emptyset$ ▷ training set

**for** $\mathbf{x} \in$ training set **do**

  $\mathbf{e}_{top} \leftarrow Quantize(E_{top}(\mathbf{x}))$

  $\mathbf{e}_{bottom} \leftarrow Quantize(E_{bottom}(\mathbf{x}, \mathbf{e}_{top}))$

  $\mathbf{T}_{top} \leftarrow \mathbf{T}_{top} \cup \mathbf{e}_{top}$

  $\mathbf{T}_{bottom} \leftarrow \mathbf{T}_{bottom} \cup \mathbf{e}_{bottom}$

**end for**

$p_{top} = \texttt{TrainPixelCNN}(\mathbf{T}_{top})$

$p_{bottom} = \texttt{TrainCondPixelCNN}(\mathbf{T}_{bottom}, \mathbf{T}_{top})$

▷ Sampling procedure

**while** true **do**

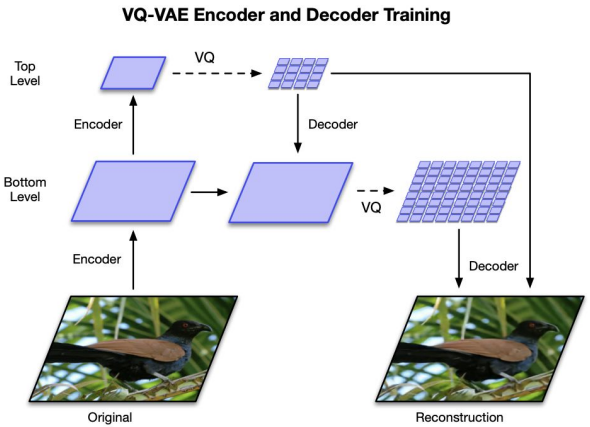
  $\mathbf{e}_{top} \sim p_{top}$

  $\mathbf{e}_{bottom} \sim p_{bottom}(\mathbf{e}_{top})$

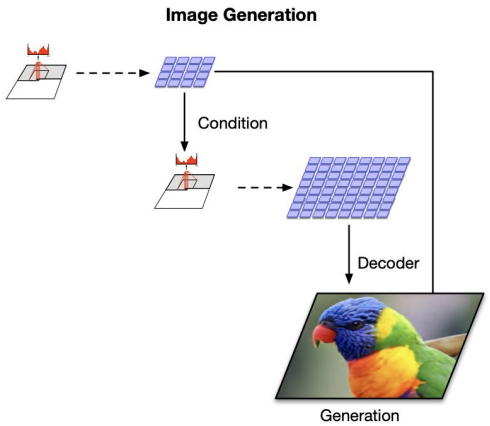
  $\mathbf{x} \leftarrow D(\mathbf{e}_{top}, \mathbf{e}_{bottom})$

**end while**

# Method (cont.)




(a) Overview of the architecture of our hierarchical VQ-VAE. The encoders and decoders consist of deep neural networks. The input to the model is a $256 \times 256$ image that is compressed to quantized latent maps of size $64 \times 64$ and $32 \times 32$ for the *bottom* and *top* levels, respectively. The decoder reconstructs the image from the two latent maps.




(b) Multi-stage image generation. The top-level PixelCNN prior is conditioned on the class label, the bottom level PixelCNN is conditioned on the class label as well as the first level code. Thanks to the feed-forward decoder, the mapping between latents to pixels is fast. (The example image with a parrot is generated with this model).

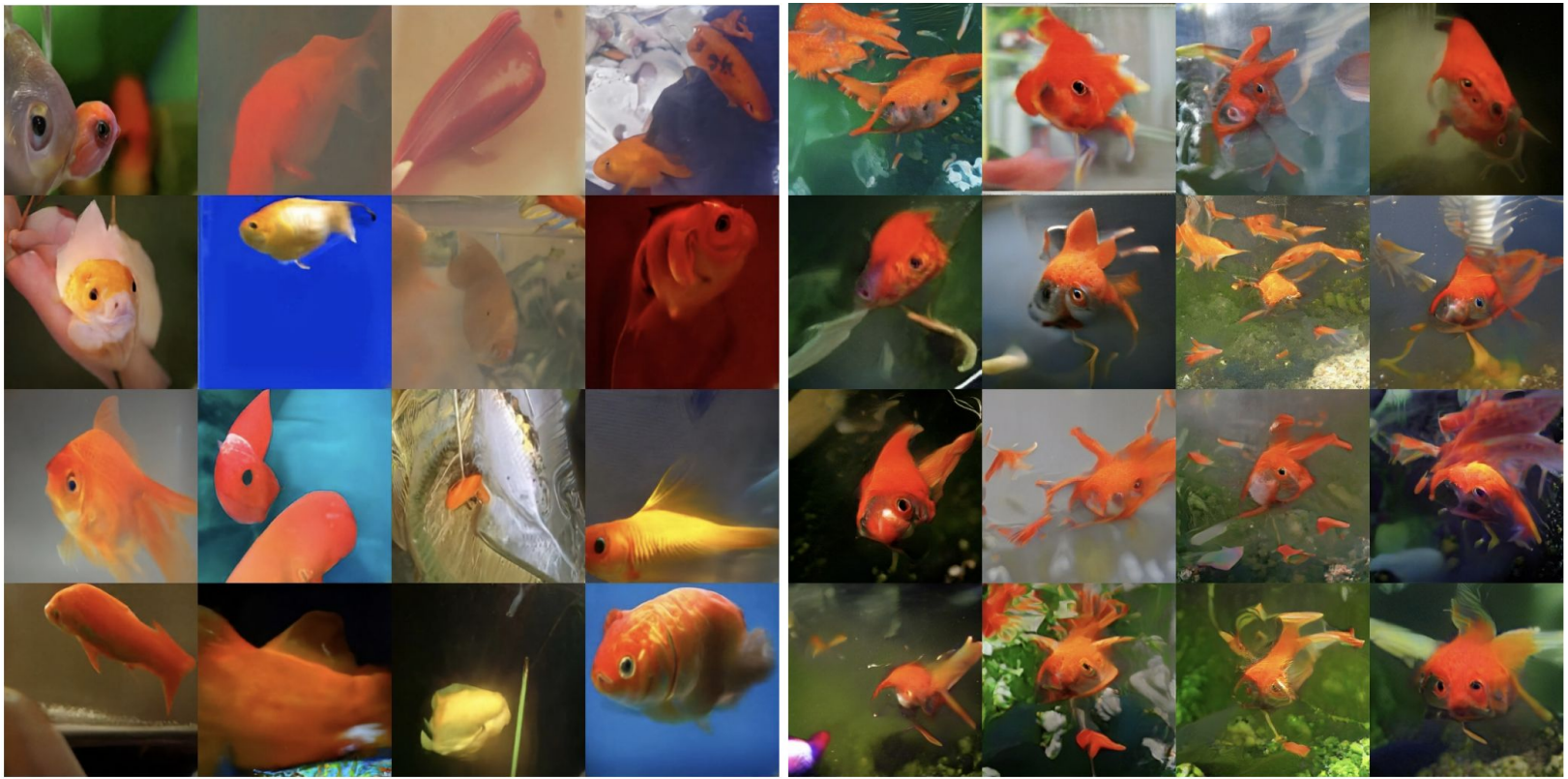
Figure 2: VQ-VAE architecture.

# Method (cont.)

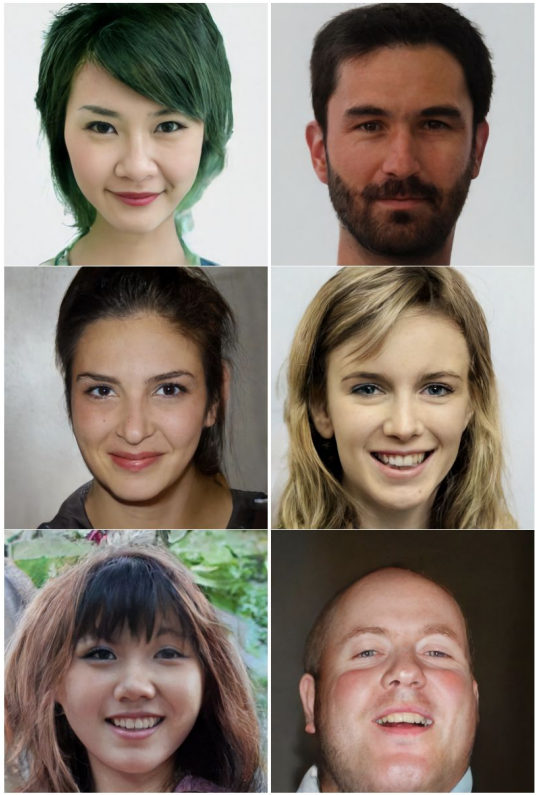

Figure 3: Reconstructions from a hierarchical VQ-VAE with three latent maps (top, middle, bottom). The rightmost image is the original. Each latent map adds extra detail to the reconstruction. These latent maps are approximately 3072x, 768x, 192x times smaller than the original image (respectively).

# Experiments

# Experiments (cont.)



# Experiments (cont.)



# Discuss

### 3.3 Trading off Diversity with Classifier Based Rejection Sampling

Unlike GANs, probabilistic models trained with the maximum likelihood objective are forced to model *all* of the training data distribution. This is because the MLE objective can be expressed as the forward KL-divergence between the data and model distributions, which would be driven to infinity if an example in the training data is assigned zero mass. While the coverage of all modes in the data distribution is an appealing property of these models, the task is considerably more difficult than adversarial modeling, since likelihood based models need to fit all the modes present in the data. Furthermore, ancestral sampling from autoregressive models can in practice induce errors that can accumulate over long sequences and result in samples with reduced quality. Recent GAN frameworks [4, 1] have proposed automated procedures for sample selection to trade-off diversity and quality. In this work, we also propose an automated method for trading off diversity and quality of samples based on the intuition that the closer our samples are to the true data manifold, the more likely they are classified to the correct class labels by a pre-trained classifier. Specifically, we use a classifier network that is trained on ImageNet to score samples from our model according to the probability the classifier assigns to the correct class.